

L3 MIAGE

Programmation orientée objet avancée (JAVA)

Cours 3 : Stream

Avant toute chose

- ▶ Les notions que nous aborderons lors des différentes séances sont des notions complexes
 - ▶ Ex : [Stream](#), livre Java I/O de O'Reilly → 479 pages !!!!!
 - ▶ Nous avons 2h de CM et 4h de TD, nous ne pourrons pas tout voir

Exceptions : Rappel

- ▶ Si une exception est susceptible d'être levée mais qu'elle n'est pas attrapée (try-catch) elle remonte à la méthode appelante et ce, jusqu'à la méthode main
- ▶ On essayera toujours d'attraper les exceptions du cas le plus particulier vers le cas le plus général
 - ▶ Ne pas se contenter de `catch(Exception e){...}`

Objectifs

- ▶ Manipuler des flux
 - ▶ Flux de caractère
 - ▶ Flux binaire
- ▶ Ouvrir, modifier, créer
 - ▶ Des fichiers txt, csv...
 - ▶ Des images
 - ▶ Des objets sérialisés

Consignes

- ▶ Pensez à l'architecture de votre code
 - ▶ Découpez en méthodes pour ne pas dupliquer votre code
 - ▶ Le main contient uniquement des objets sur lesquels sont appelées des méthodes
 - ▶ Pas de boucles, de déclaration de fonction dans le main et la classe principale
- ▶ Si vous utilisez un IDE et que vous travaillez dans un projet, vous devriez avoir un dossier src contenant vos fichiers .java
 - ▶ Au même niveau que le dossier src, créez un fichier data
 - ▶ Dans ce fichier data, créez un dossier input et un dossier output
 - ▶ Tout fichier lu doit se trouver dans le dossier input, et tout fichier créé doit être dans le dossier output

Lecture d'un fichier

- ▶ Téléchargez sur Arche le fichier movies.csv
- ▶ Ouvrez-le avec un tableur et créez la classe vous permettant de représenter les informations contenues dans ce fichier
- ▶ Créez quelques méthodes (présence d'un mot dans le titre, savoir si le film est sorti avant ou après votre naissance...)
- ▶ Utilisez la méthode adaptée pour lire ce fichier ligne par ligne et créez vos objets en conséquence
- ▶ Les objets seront stockés dans un tableau

Ecriture dans un fichier

- ▶ A partir des objets que vous avez créé et qui sont stockés dans un tableau
 - ▶ Ouvrez un flux de caractères
 - ▶ Parcourez votre tableau
 - ▶ Créez une nouvelle ligne avec les attributs séparés par le caractère de votre choix
 - ▶ Fermez le flux de caractères
- ▶ Vérifiez que le fichier généré correspond bien à vos attentes
 - ▶ Pour en être sûr, essayez de l'ouvrir de nouveau avec la méthode que vous avez créée pour lire le fichier initial
- ▶ Félicitations, vous venez de sauvegarder vos objets

Pratique ?

- ▶ Imaginez maintenant que vous n'avez pas une mais des dizaines de classes qui sont en plus imbriquées les unes dans les autres
 - ▶ Pensez-vous que cette solution soit viable ?
- ▶ Quel autre procédé pouvez-vous utiliser afin de sauvegarder vos objets ?
 - ▶ La sérialisation (que vous avez dû voir dans le poly de cours)
- ▶ Modifiez votre code afin de rendre vos données sérialisables et sauvegardez les dans un fichier avec l'extension .dat par exemple

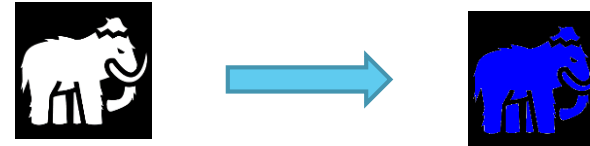
Sérialisation

- ▶ Une fois que vous avez sauvegardé vos objets dans un fichier .dat
 - ▶ Modifiez votre classe (ajoutez/supprimez un attribut, modifiez une méthode...)
 - ▶ Essayez de désérialiser votre fichier .dat
 - ▶ Que constatez-vous ?
 - ▶ Quelle en est la raison ?
 - ▶ Comment pouvez-vous éviter ce genre d'incident ?

Sérialisation et flux de caractères

- ▶ La dé/sérialisation est-elle plus ou moins rapide que la lecture/écriture des objets via un flux de caractère ?
 - ▶ Ecrivez le code nécessaire permettant de connaître le réponse
- ▶ Lorsque vous souhaitez sauvegarder vos objets, quels sont les avantages et les inconvénients de chacune des deux méthodes ?

Manipulation d'images : échauffement



- ▶ Téléchargez l'image “mamouth.png” sur Arche
- ▶ Le but de cet exercice est de changer la couleur de tous les pixels blancs par des pixels bleus et de sauvegarder l'image dans le dossier output sous le nom “mamouth_bleu.png”
- ▶ Ne vous contentez pas de faire du copier coller avec ce que vous trouverez sur les forums/tuto
 - ▶ Ne cherchez pas la solution directement mais les étapes intermédiaires
 - ▶ Comment ouvrir une image ?
 - ▶ Comment accéder à la couleur d'un pixel ?
 - ▶ ...
 - ▶ Prenez le temps de comprendre quels sont les imports que vous faites

Manipulation d'images : pour de vrai

- ▶ Téléchargez l'image Sherlock sur l'ENT
- ▶ Ouvrez cette image pixel par pixel afin de créer une nouvelle image que vous créerez comme suit :
 - ▶ Si le dernier bit de la couleur (Rouge, bleu ou vert, les trois ont la même valeur) d'un pixel convertit en binaire est égal 0
 - ▶ Alors le pixel correspondant dans la nouvelle image sera blanc
 - ▶ Sinon il sera noir



?

Manipulation d'images : pour de vrai

- ▶ Vous avez fini ? Créez maintenant le code permettant de faire l'inverse de ce que vous venez de réaliser
- ▶ Vous souhaitez aller plus loin
 - ▶ Utilisez des niveaux de gris et non plus une image en noir et blanc uniquement. Un niveau de gris est codé sur un octet, c'est-à-dire une valeur entre 0 et 255. Il faudra donc réfléchir à la manière dont vous pourrez réaliser cette modification.